

**Original citation:**

Beynon, Meurig and Buckle, J. F. (1985) Computation equivalence and replaceability in finite algebras. University of Warwick. Department of Computer Science. (Department of Computer Science Research Report). (Unpublished) CS-RR-072

**Permanent WRAP url:**

<http://wrap.warwick.ac.uk/60771>

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**A note on versions:**

The version presented in WRAP is the published version or, version of record, and may be cited as it appears here. For more information, please contact the WRAP Team at: [publications@warwick.ac.uk](mailto:publications@warwick.ac.uk)



<http://wrap.warwick.ac.uk/>

The University of Warwick

THEORY OF COMPUTATION

REPORT No. 72

Computational Equivalence  
and Replaceability in  
Finite Algebras

by

W. M. Beynon

J. F. Buckle

Department of Computer Science  
University of Warwick  
Coventry CV4 7AL  
England

August 1985

# Computational equivalence and replaceability in finite algebras

*Meurig Beynon and John Buckle*

Dept. of Computer Science,  
University of Warwick,  
Coventry CV4 7AL

## ABSTRACT

In this paper, we consider computational equivalence and replaceability in various classes of finite algebras, and show in particular how to characterise these relations for finite lattices and semi-lattices. Results include generalisations of theorems previously proved for distributive lattices, and a description of computational equivalence and replaceability in the multiplicative semigroup of residues modulo an integer. Non-replaceability of monotone boolean functions is shown to be NP-complete.

August 29, 1985

## Introduction.

The algebraic study of replaceability and computational equivalence relations on finite distributive lattices was initiated in [1] and [2], with the analysis of monotone boolean circuits in mind. As explained in [2], replaceability and computational equivalence are meaningful concepts in a general algebraic setting, and our purpose in this paper is to explore generalisations, extensions and analogues of the results of [2] in several different algebraic contexts.

Though the results described and illustrated below are of an abstract algebraic nature, and possibly have no direct computational application, they help to put the ideas considered in [2] into a wider perspective, and are motivated by computational considerations in a broad sense. The study of monotone boolean circuits has shown how replaceability can be helpful both as a tool for developing new algorithms and for proving lower bounds (cf. [3],[7],[11],[12]), and it is reasonable to envisage similar applications in other areas. The fact that some aspects of computational equivalence for semigroups have already been studied in connection with the theory of syntactic monoids (see e.g. [9], [10] and [13]) suggests, at the very least, that applications can be profitably sought elsewhere.

As mentioned in [2], computational equivalence is uninteresting for algebras such as groups in which congruences with singleton equivalence classes are trivial, and this may be seen as limiting the scope for useful application. In view of this, it is worth emphasising that the most appropriate setting for studying a class of computations may in practice be an algebra with less structure than it might at first appear. For instance, whilst it may be helpful to regard the computation of a monotone boolean function  $f(x_1, x_2, \dots, x_n)$  as the computation of an element in the free distributive lattice  $\text{FDL}(n)$ , it can be unrealistic in as much as the word problem in  $\text{FDL}(n)$  ("given monotone formula  $g(x_1, x_2, \dots, x_n)$  and  $h(x_1, x_2, \dots, x_n)$ , do  $g$  and  $h$  represent the same element of  $\text{FDL}(n)$ ?" ) is NP-hard (cf. §3 ). In effect, computations take place within an algebra of representations of elements of  $\text{FDL}(n)$  by monotone boolean formula which can possibly be modelled more faithfully by an alternative algebraic structure.

The paper is in five sections. In §1, we give essential preliminaries. In §'s 2 and 3, we consider how far the results in [2] can be generalised to arbitrary finite lattices, and describe two different methods by which replaceability and computational equivalence relations can be determined. In §3, we describe a simple algorithm which decides the validity of a given replaceability relation between elements of a finite lattice  $L$  in time polynomial in the size of the lattice, and show that non-replaceability of monotone boolean functions is NP-complete. In §5, we consider some aspects of computational equivalence and replaceability in particular finite commutative semigroups. As a corollary, we obtain a curious characterisation of replaceability in a finite distributive lattice in terms of replaceability in the associated meet- and join-semilattices.

The proofs in §'s 2 and 3 are due to John Buckle.

### §1. Replaceability and computational equivalence.

For convenience, we include a few basic definitions and results on replaceability and computational equivalence. For more details, see [2].

Suppose that  $A$  is an  $\Omega$ -algebra generated by  $a_1, a_2, \dots, a_n$ , and that  $F \subseteq A$ . A preorder relation  $\sqsubseteq_F$  associated with  $F$  is defined by  $h \sqsubseteq_F g$  (also written  $g \supseteq_F h$ , and read:  $h$  may replace  $g$  with respect to computation of  $F$ ) if for all  $f$  in  $F$ :

"if  $w$  is an  $\Omega$ -word such that  $w(g, a_1, a_2, \dots, a_n) = f$ , then  $w(h, a_1, a_2, \dots, a_n) = f$ ".

That is, for each  $f$  in  $F$ , an  $\Omega$ -formula over  $A$  which represents  $f$  in terms of  $g$  still represents  $f$  when  $g$  is replaced by  $h$ .

The elements  $g$  and  $h$  are *computationally equivalent* modulo  $F$  (written  $g \sqsubseteq_F h$ ) iff

$$g \sqsubseteq_F h \text{ and } h \sqsubseteq_F g;$$

the relation  $\sqsubseteq_F$  then defines a partial order on the computational equivalence classes of  $\sqsubseteq_F$ . If  $F = \{f\}$ , it will be convenient to write  $\sqsubseteq_f$  for  $\sqsubseteq_F$ , and  $\square_f$  for  $\square_F$ .

#### Lemma 1.1.

If  $f \in F \subseteq A$ , then  $\sqsubseteq_f$  respects the operations in  $\Omega$  on  $A$ :

if  $\omega \in \Omega$  has arity  $k$ , and  $g_i \sqsubseteq_f h_i$  for  $1 \leq i \leq k$ , then

$$\omega(g_1, g_2, \dots, g_k) \sqsubseteq_f \omega(h_1, h_2, \dots, h_k),$$

and  $\sqsubseteq_F = \bigcap \{ \sqsubseteq_f \mid f \in F \}$ .

Moreover:  $\square_f$  is an  $\Omega$ -congruence on  $A$ , and  $\square_F = \bigcap \{ \square_f \mid f \in F \}$ . ■

If  $\alpha$  is an equivalence relation on a set  $S$ , and the equivalence class of  $s$  contains a single element,  $s$  will be called *solitary under  $\alpha$* , or simply *solitary* where the equivalence relation is clear from the context.

#### Lemma 1.2.

$\sqsubseteq_F$  is the unique maximal preorder relation on  $A$  respecting the operations in  $\Omega$  such that all elements in  $F$  are minimal (i.e: if  $f \in F$  and  $g \sqsubseteq_F f$ , then  $g = f$ .)

$\square_F$  is the unique maximal  $\Omega$ -congruence on  $A$  such that each  $f$  in  $F$  is solitary. ■

Lemma 1.2 shows that computational equivalence is a trivial relation in the context of many choices of  $\Omega$ . For instance, if  $A$  is a group or ring, a congruence class will be a coset of a subgroup containing 2 or more elements in a non-trivial case. However, where computational equivalence is trivial, the preorder by replaceability may still be of interest. This is the case, for example, in boolean lattices (cf. [2]).

In view of Lemmas 1.1 and 1.2, it will suffice to consider computational equivalence and replaceability modulo a single element. The above observations then suggest some natural questions. For a given class of algebras, it is of

interest to determine general principles for computing replaceability and computational equivalence relations, and to seek characteristic properties of the partially ordered algebraic quotients which result. A detailed theory of this type has been developed for finite distributive lattices, and is described in [2]. Our interest here is in presenting some preliminary results on other classes of algebras for which a non-trivial theory can be developed, viz. finite lattices and certain commutative semigroups.

## §2. Computational equivalence in finite lattices in terms of join-irreducibles.

In this section, computational equivalence and replaceability in finite lattices is described in terms of sets of join-irreducibles. The results in this section generalise theorems relating to distributive lattices proved in [2], in particular Cor. 3.4. (For lattice theoretic background, see [5] or [8]).

### Definitions.

If  $L$  is a finite lattice and  $f \in L$ , then  $P(f)$  and  $Q(f)$  are defined as

$$P(f) = \{ p \text{ is a join-irreducible} \mid \exists u < f : p \text{ is minimal subject } u \vee p = f \}$$

$$Q(f) = \{ q \text{ is a meet-irreducible} \mid \exists u > f : q \text{ is maximal subject } u \wedge q = f \}$$

By definition,  $\forall p \in P(f) \ \& \ \forall x < p : p \not\leq x$ , because  $p$  is minimal such that  $p \vee u = f$  for some  $u \in L$ . Similarly,  $\forall q \in Q(f) \ \& \ \forall x > q : q \not\leq x$ .

Lemma 2.1:

Let  $f \in L$ , a finite lattice, then  $f = \bigvee P(f) = \bigwedge Q(f)$

Proof:

Choose  $P$  such that  $\bigvee P = f$  is an irredundant representation for  $f$  as a join of join-irreducibles. The lemma is proved by defining a sequence of sets of join-irreducibles  $P^0 = P, P^1, \dots, P^k$  such that  $f = \bigvee P^i$  irredundantly for  $i \geq 0$ , and  $P^k \subset P(f)$ .

Suppose that  $P^0, P^1, \dots, P^i$  have been defined, and that  $P^i \not\subset P(f)$ . Then for some  $p$  in  $P^i$ ,  $\exists z < p$  such that  $z \vee \bigvee (P^i \setminus \{p\}) = f$ . Let  $z = \bigvee P'$  be an irredundant representation for  $z$  as a join of join-irreducibles, and define  $P^{i+1}$  to be a subset of  $P' \cup P^i \setminus \{p\}$  such that  $f = \bigvee P^{i+1}$  irredundantly. Only a finite sequence of subsets  $P^0, P^1, \dots, P^k$  can be generated in this way, since chains of join-irreducibles in a finite lattice have finite length.

A similar argument is used to prove  $f = \bigwedge Q(f)$ .

Define  $P_f$  to be the set of maximal elements of  $P(f)$ , and  $Q_f$  to be the set of minimal elements of  $Q(f)$ . Obviously  $\bigvee P_f = \bigvee P(f) = f$ . The definition of the set  $P_f$  coincides with the definition of prime implicants of a function when the lattice is free distributive. If the lattice is modular, then  $P_f$  can be alternatively defined as

$$P'_f = \{ p \mid p \text{ is join-irreducible and maximal subject to } \exists u < f : u \vee p = f \}$$

To see that the definitions are equivalent it will suffice to show that no element less than  $p \in P_f$  can replace  $p$ . Suppose  $x \leq p$  and  $p \sqsupset_f x$ ; since  $p \in P_f$  there exists  $u < f$  such that  $u \vee p = f$ . Hence  $x \vee u = f$  because  $p \sqsupset_f x$ . Therefore  $p = p \wedge f = p \wedge (x \vee u) = (p \wedge u) \vee x$  by modularity ( $x \leq p$ ). However  $p \wedge u < p$  and  $p$  is a join-irreducible, hence  $x = p$ .

For any join-irreducible  $p$  and meet-irreducible  $q$ :

$$\tilde{p} = \{ q \mid q \text{ is maximal subject to } q \not\leq p \}$$

$$\tilde{q} = \{ p \mid p \text{ is minimal subject to } p \not\leq q \}$$

Obviously  $\tilde{p}$  is a set of meet-irreducibles and  $\tilde{q}$  is a set of join-irreducibles. Also for any element  $x$ ,  $p \not\leq x$  if and only if  $\exists q \in \tilde{p}: x \leq q$  and dually. If  $F$  is a set of irreducibles, then  $\tilde{F}$  will be used to denote  $\bigcup_{x \in F} \tilde{x}$ , and  $\tilde{F}^2 = \{ f \in \tilde{F} \mid g \in \tilde{F} \}$ . Let  $F$  be

a set of join-irreducible, define a sequence of sets of join-irreducibles  $F_0, F_1, \dots, F_k$  via  $F_0 = F$ ,  $F_{i+1} = F_i \cup \tilde{F}_i^2$ . Let  $F^* = F_k$  where  $F_{k+1} = F_k$ . (This is bound to occur since there are only a finite number of join-irreducibles and the sets are non decreasing.)

For any set of join-irreducibles  $F$ , let  $F[x] = \{ y \in F \mid y \leq x \}$ .

The next proposition shows that in a modular lattice the  $\sim$  function is reflexive in that  $p \in \tilde{q} \Leftrightarrow q \in \tilde{p}$ . Hence  $\tilde{F}^2 \supset F$  and  $F^* = \tilde{F}^k$  for some  $k$ .

Proposition 2.2:

If  $M$  is a modular lattice and  $p$  a join-irreducible and  $q$  a meet-irreducible in  $M$ , then

$$p \in \tilde{q} \text{ if and only if } q \in \tilde{p}.$$

Proof:

Suppose  $p \in \tilde{q}$ , then  $p$  is a minimal element  $\not\leq q$ , hence  $p > p \wedge q$ . To prove  $q \in \tilde{p}$  it will suffice to show that  $p \vee q > q$ , hence  $q$  is a maximal element  $\not\leq p$ .

Choose  $x$  such that  $p \vee q > x$  and  $x \geq q$ . Then by modularity  $x = x \wedge (q \vee p) = q \vee (x \wedge p) = q$ . Hence  $x = q$  and  $p \vee q > q$ .

The case for  $q \in \tilde{p}$  is similar. ■

Let  $>_f$  be a pre-order defined on a finite lattice  $L$  by:

$$g >_f h \Leftrightarrow P_f^*[g] \subset P_f^*[h] \text{ and } \tilde{Q}_f^*[g] \supset \tilde{Q}_f^*[h]$$

where  $f, g, h \in L$ .

Lemma 2.3:

The pre-order  $>_f$  respects  $\wedge$  and  $\vee$  on a finite lattice  $L$ , ie if  $g, h \in L$ :

$$(\forall a \in L) \ g >_f h \Rightarrow (a \wedge g >_f a \wedge h \text{ and } a \vee g >_f a \vee h)$$

Proof:

$>_f$  respects  $\wedge$ : Assume  $g >_f h$ : then  $P_f^*[g] \subset P_f^*[h]$  and  $\tilde{Q}_f^*[g] \supset \tilde{Q}_f^*[h]$ . Since

$$P_f^*[g \wedge a] = P_f^*[a] \cap P_f^*[g] \text{ and } P_f^*[g] \subset P_f^*[h]$$

it follows that  $P_f^*[g \wedge a] \subset P_f^*[a] \cap P_f^*[h] = P_f^*[a \wedge h]$ . A similar argument shows that

$$\tilde{Q}_f^*[h \wedge a] \subseteq \tilde{Q}_f^*[g \wedge a].$$

$\succ_f$  respects  $\vee$ . It will suffice to show that if  $g \vee a \not\succ_f h \vee a$ , then  $g \not\succ_f h$ . Suppose  $\exists x \in P_f^*$  such that  $x \leq a \vee g$  but  $x \not\leq a \vee h$ , then  $\exists w \in X$  such that  $w \geq a \vee h$  and  $w \not\leq a \vee g$ . Since  $w \not\leq a \vee g$  and  $w \geq a$  it follows that  $w \not\leq g$ . Thus  $\exists y \in W$  such that  $g \geq y$ , and  $h \not\geq y$  as  $h \leq w$ . Thus  $P_f^*[g] \not\subseteq P_f^*[h]$  since  $y \in X \subseteq P_f^*$ .

The argument for  $x \leq a \vee h$  and  $x \not\leq a \vee g$  follows a similar method using the  $\tilde{Q}_f^*$  set.

Theorem 2.4:

If  $f \in L$ , a finite lattice, then  $\succ_f = \sqsubset_f$

Proof:

By Lemma 1.2 it will suffice to show that  $\succ_f \supseteq \sqsubset_f$  and that  $f$  is minimal in that  $f \succ_f g \Leftrightarrow g = f$ .

Suppose that  $f \succ_f g$ . If  $p \in P_f$ , then  $p \in P_f^*[f] \subseteq P_f^*[g]$ , hence  $p \leq g$ . Thus  $f \leq g$  since  $f = \vee P_f$ . If  $p' \notin \tilde{Q}_f$ , then  $p' \notin \tilde{Q}_f^*[f] \supseteq \tilde{Q}_f^*[g]$  hence  $p' \not\leq g$ . Thus  $\forall q \in Q_f: q \geq g$  and  $g \leq f$ .

To complete the proof it will be enough to show that  $g \not\succ_f h$  implies  $g \not\sqsubset_f h$ . Suppose then that  $g \not\succ_f h$ . There are two possible cases:

Case 1:  $\exists p \in P_f^*$  such that  $g \geq p$  and  $h \not\geq p$

Case 2:  $\exists p \in \tilde{Q}_f^*$  such that  $g \not\geq p$  and  $h \geq p$

Case 1. Since  $p \in P_f^*$  there exists a sequence of join and meet irreducibles  $p = p_0, q_1, p_2, q_3, \dots, p_i \in P_f$  such that  $p_j \in \tilde{Q}_{j+1}$  and  $q_j \in \tilde{P}_{j+1}$ , for  $1 \leq j \leq i$ . Define  $w(x)$  to be the lattice word

$$w(x) = (\dots (((x \wedge p) \vee q_1) \wedge p_2) \vee \dots \vee q_{i-1}) \wedge p_i$$

From the definition of  $P_f^*$  and the argument below, which shows  $w(g) = p_i$  and  $w(h) < p_i$ , it follows that  $g \not\sqsubset_f h$ . Since  $g \geq p$  and  $p \not\leq q_1$  it follows that  $(g \wedge p) \vee q_1 > q_1$ . Moreover  $q_1$  is maximal subject to  $q_1 \not\leq p_2$  because  $q_1 \in \tilde{P}_2$ , hence  $((g \wedge p) \vee q_1) \wedge p_2 = p_2$ . Thus

$$w(g) = (\dots ((p_2 \vee q_3) \wedge p_4) \vee \dots \vee q_{i-1}) \wedge p_i$$

It follows by induction that  $w(g) = p_i$ . On the other hand  $(h \wedge p) \vee q_1 = q_1$  and  $(q_1 \wedge p_2) < q_3$  so  $(q_1 \wedge p_2) \vee q_3 = q_3$  and

$$w(h) = (\dots (q_3 \wedge p_4) \vee \dots \vee q_{i-1}) \wedge p_i$$

which by induction equals  $w(h) = q_{i-1} \wedge p_i < p_i$

Case 2. In this case let

$$v(x) = (\dots (((x \wedge p) \vee q_1) \wedge p_2) \vee \dots) \vee q_{i-1}$$

where  $q_1, p_2, \dots, q_{i-1} \in Q_f$  is a sequence of irreducibles such that  $p_j \in \tilde{Q}_{j-1}$  and  $q_j \in \tilde{P}_{j-1}$ , for  $1 \leq j \leq i$ . By adapting the argument above it can be shown that  $v(h) > q_i$  and  $v(g) = q_i$  hence  $g \not\sqsubset_f h$ .



As an immediate corollary to Theorem 2.4:

Cor. 2.5:

$$g \sqsubset_f h \text{ iff } P_f^*[g] = P_f^*[h] \text{ and } \tilde{Q}_f^*[g] = \tilde{Q}_f^*[h].$$

Theorem 2.4 and Cor. 2.5 generalise results previously established for finite distributive lattices (c.f. [2] Cor. 3.4), but other results described in [2] cannot be generalised. For instance, computational equivalence is not in general defined by a retract of a finite lattice onto an interval. It is of interest to note however that all preorders and congruences on a finite lattice which respect the lattice operations can be defined in a fashion similar to that described above (see [4] and compare [2] Lemma 2.1).

### §3. Computational equivalence and replaceability in terms of covering edges.

In this section, an alternative characterisation of  $\sqsubset_f$  and  $\sqsubset_f$  is described. If  $a, b$  are two elements of a finite lattice, then  $a \succ b$  ( $a$  covers  $b$ ) if  $\forall x \leq a : x \succ b \Rightarrow x = a$ . When two elements have a covering relationship between them, they are connected by a edge in the Hasse diagram. In this section it is proved that elements of a finite lattice are computationally equivalent or replaceable if they are connected by a path of special covering edges.

The first part of this section defines a relation between covering edges in the lattice from which a pre-order " $\prec$ " is defined. This pre-order is then shown to be equivalent to replaceability. This leads in particular to an alternative definition of the sets  $P_f^*$  and  $\tilde{Q}_f^*$  of the previous section.

**Defn** If  $L$  is a finite lattice,  $a, b \in L$ , and  $a \succ b$ , then the pair  $(a, b)$  is called a **covering edge** and denote by  $\langle a, b \rangle$ .

A lattice word  $w$  is an **alternating word** it can be expressed as  $w(x) = (\dots ((x \wedge z_1) \vee z_2) \wedge \dots) \wedge z_n$  where  $n \geq 0$  and  $z_1, z_2, \dots, z_n$  are lattice elements. (This definition includes alternating words beginning with  $\vee$  since it is possible to set  $z_1 = 1$ .)

If  $\langle a, b \rangle$  and  $\langle c, d \rangle$  are covering edges, and  $w$  is an alternating word such that  $w(a) = c$  and  $w(b) = d$ , then  $\langle a, b \rangle$  reduces to  $\langle c, d \rangle$  (denoted  $\langle a, b \rangle \rightarrow \langle c, d \rangle$ ) via  $w$ .

Obviously  $\rightarrow$  is a reflexive and transitive relation, however it can be easily shown that in a general lattice that it is not symmetric (see Example 1). If  $L$  is a modular lattice, then  $\rightarrow$  is an equivalence relation.

**Proposition 3.1:**

In a modular lattice  $M$ , the relation  $\rightarrow$  defines an equivalence relation on the covering edges of  $M$ .

**Proof:**

Let  $a, b \in M$  such that  $a \succ b$  and let  $c \in M$  be any element. It will be shown that either  $a \vee c \succ b \vee c$  or  $a \vee c = b \vee c$  and dually. Let  $d$  be any element such that  $a \vee c \geq d \geq b \vee c$ , then  $d = d \wedge (a \vee b \vee c)$  and by modularity  $d \wedge (a \vee (b \vee c)) = (b \vee c) \vee (d \wedge a)$ . If  $d \geq a$  then  $d = (b \vee c) \vee (d \wedge a) = a \vee c$ , otherwise  $d \wedge a < a$  so  $d \wedge a < b$  and

$d = (b \vee c) \vee (d \wedge a) = b \vee c$ . Similarly for  $\wedge$ .

To prove that  $\langle a, b \rangle \rightarrow \langle c, d \rangle$  implies  $\langle c, d \rangle \rightarrow \langle a, b \rangle$  for two covering edges  $\langle a, b \rangle$  and  $\langle c, d \rangle$ , induction on the length of the alternating word mapping  $\langle a, b \rangle$  to  $\langle c, d \rangle$  is used.

*Base case.* Suppose that  $\langle a, b \rangle \rightarrow \langle c, d \rangle$  via a word  $w(x)$  of length one, ie  $w(x) = x \vee y$  or  $w(x) = x \wedge y$  for some element  $y \in M$ . If  $w(x) = x \vee y$  then the lattice word  $v(x) = x \wedge a$  will map  $\langle c, d \rangle \rightarrow \langle a, b \rangle$  because  $c \wedge a = a$  and  $b \leq d \wedge a \wedge a$ . Similarly if  $w(x) = x \wedge y$  then  $v(x) = x \vee b$  is an inverse map.

*Induction Step.* Let  $\langle a, b \rangle \rightarrow \langle c, d \rangle$  by an alternating word  $w(x)$  of length  $n$ . In this case the  $w(x)$  can be split into two words  $u(x)$  and  $v(x)$  of lengths  $n-1$  and 1. By the first part of the proof  $\langle u(a), u(b) \rangle$  is a covering edge, hence by induction there exists an inverse word  $u'(x)$  mapping  $\langle u(a), u(b) \rangle \rightarrow \langle a, b \rangle$  and a word  $v'(x)$  mapping  $\langle c, d \rangle \rightarrow \langle u(a), u(b) \rangle$ . Hence by transitivity there is an alternating word  $w'(x)$  mapping  $\langle c, d \rangle \rightarrow \langle a, b \rangle$ . ■

Lemma 3.2:

Suppose that  $L$  is a finite lattice,  $c \in L$  and  $\langle a, b \rangle, \langle g, h \rangle$  are covering edges in  $L$ .

If there exists  $x, y$  such that  $a \vee c \geq x > y \geq b \vee c$  and  $\langle x, y \rangle \rightarrow \langle g, h \rangle$  then  $\langle a, b \rangle \rightarrow \langle g, h \rangle$ , and dually.

Proof:

Let  $w$  be the alternating word  $w(z) = ((z \vee y) \wedge x)$ . Then  $w(a) = x$  and  $w(b) = y$ , hence  $\langle a, b \rangle \rightarrow \langle x, y \rangle$ , and by transitivity of  $\rightarrow$  the result is proved. ■

**Defn** Let  $L$  be a finite lattice and let  $a, b, f \in L$  such that  $a > b$ . If  $\langle a, b \rangle \rightarrow \langle f, z \rangle$  for some  $z \in L$  such that  $f > z$  then  $\langle a, b \rangle$  is a **non-upper-replaceable-edge with respect to  $f$**  (ie  $b$  can't replace  $a$ ). If there doesn't exist such an alternating word then  $\langle a, b \rangle$  is a **upper-replaceable-edge with respect to  $f$** .

If  $\langle a, b \rangle \rightarrow \langle y, f \rangle$  for some  $y \in L$  such that  $y > f$  then  $\langle a, b \rangle$  is a **non-lower-replaceable-edge with respect to  $f$** . In the remainder of this section only replaceable edges with respect to  $f$  are considered, and the "wrt  $f$ " clause is omitted.

Define a pre-order  $\leq_f$  on  $L$  such that  $a \leq_f b$  if  $\exists n \geq 0$  and a sequence of elements  $x_0 (=a), x_1, x_2, \dots, x_n (=b)$  where *either*

$x_i > x_{i+1}$  and  $\langle x_i, x_{i+1} \rangle$  is a lower-replaceable-edge

*or*

$x_{i+1} > x_i$  and  $\langle x_{i+1}, x_i \rangle$  is an upper-replaceable-edge

Lemma 3.3:

If  $L$  is a finite lattice and  $f \in L$ , then the pre-order  $\leq_f$  respects the lattice operations.

Proof:

Suppose  $x \leq_f y$ . It will be shown that  $\forall a: x \wedge a \leq_f y \wedge a$  and  $x \vee a \leq_f y \vee a$ .

Assume  $x \wedge a \neq y \wedge a$ . Since  $x \leq_f y$  there exists a sequence of elements  $z_0 (=x), z_1, \dots, z_k (=y)$  such that *either*  $z_i > z_{i+1}$  and  $\langle z_i, z_{i+1} \rangle \vdash \langle y, f \rangle$  for some  $y > f$  or  $z_{i+1} > z_i$  and  $\langle z_{i+1}, z_i \rangle \vdash \langle f, z \rangle$  for some  $z$  where  $f > z$ .

Consider the sequence  $z_0 \wedge a, z_1 \wedge a, \dots, z_k \wedge a$ . If  $\exists i$  such that  $z_i \wedge a = z_{i+1} \wedge a$  then remove  $z_{i+1} \wedge a$  from the sequence. If  $z_i \wedge a > z_{i+1} \wedge a$  but does not cover it then introduce new elements so that there is a covering chain from  $z_i \wedge a$  to  $z_{i+1} \wedge a$ . By Lemma 3.2 every edge between  $z_i \wedge a$  and  $z_{i+1} \wedge a$  in this covering chain is lower-replaceable since  $\langle z_i, z_{i+1} \rangle$  is lower-replaceable. Similarly introduce new elements if necessary for  $z_{i+1} \wedge a > z_i \wedge a$ . Hence there exists a sequence of elements from  $x \wedge a$  to  $y \wedge a$  such that  $x \wedge a \leq_f y \wedge a$ .

The case for  $\vee$  is similar.

Theorem 3.4:

Let  $L$  be a finite lattice and let  $a, b \in L$ , then

$$a \leq_f b \Leftrightarrow a \sqsubseteq_f b.$$

Proof:

By Lemma 1.2 and Lemma 3.3 above it will suffice to show that  $\leq_f$  contains  $\sqsubseteq_f$  and  $f$  is minimal under  $\leq_f$  (ie  $g \leq_f f \Rightarrow g=f$ ).

By definition of  $\leq_f$ ,  $f$  is minimal since no edge adjacent to  $f$  is lower or upper-replaceable.

Let  $a, b \in L$  be such that  $a \not\leq_f b$ ; it will be shown that  $a \not\sqsubseteq_f b$ . Consider a sequence of elements  $a=x_0, x_1, \dots, x_k=a \wedge b$  and  $b=y_0, y_1, \dots, y_m=a \vee b$ , where  $x_i > x_{i+1}$  and  $y_i > y_{i+1}$ . Since  $a \not\leq_f b$  there exists *either* a non-lower-replaceable-edge  $\langle x_i, x_{i+1} \rangle$  or a non-upper-replaceable-edge  $\langle y_i, y_{i+1} \rangle$ .

In the former case, since  $\langle x_i, x_{i+1} \rangle \vdash \langle y, f \rangle$  for some  $y > f$ , there exists an alternating word  $w(x)$  such that  $w(x_i)=y$  and  $w(x_{i+1})=f$ . Hence the lattice word  $v(x)=w((x \wedge x_i) \vee x_{i+1})$  maps  $a$  to  $y$  and  $b$  to  $f$  and  $a \not\sqsubseteq_f b$ .

The latter case is dealt with similarly.

**Defn** Let  $L$  be a finite lattice and let  $a, b \in L$  such that  $a > b$ . If  $\langle a, b \rangle$  is both a upper-replaceable and a lower-replaceable-edge then  $\langle a, b \rangle$  is a **collapsible** edge.

Define an equivalence relation  $\sim_f$  by  $a \sim_f b$  if  $a=b$  or there exists a path of collapsible edges from  $a$  to  $b$ .

Theorem 3.5:

Let  $L$  be a finite lattice and let  $a, b \in L$ , then

$$a \sim_f b \Leftrightarrow a \sqsubseteq_f b$$

Proof:

By Lemma 1.2 it will suffice to show that  $\square_f$  is a lattice congruence which leaves  $f$  solitary and contains  $\square_f$ .

The proof that  $\square_f$  is a congruence is similar to the proof of Lemma 3.3, and uses the same construction for the new sequence from  $x \wedge a$  to  $y \wedge a$  and from  $x \vee a$  to  $y \vee a$ .

Obviously  $\square_f$  leaves  $f$  solitary since no edges adjacent to  $f$  is collapsible.

Lastly, the proof that  $\leq_f$  contains  $\square_f$  in Theorem 3.4 can be adapted to prove that  $\square_f$  contains  $\square_f$ , and is left to the reader.

An alternative definition of the sets  $P_f^*$  and  $Q_f^*$  can now be given.

Defn Let  $f \in L$ , a finite lattice, and let  $P$  be the set of all join-irreducibles,

$P_f^* = \{p \in P \mid \langle p, k \rangle \text{ is a covering edge and is non-upper-replaceable}\}$

$Q_f^* = \{p \in P \mid \langle p, k \rangle \text{ is a covering edge and is non-lower-replaceable}\}$

Theorem 3.6:

Let  $f, g, h$  be elements of a finite lattice  $L$ , then

$$g \sqsubset_f h \Leftrightarrow P_f^*[g] \supset P_f^*[h] \text{ and } Q_f^*[g] \subseteq Q_f^*[h]$$

Proof:

$\Rightarrow$ : Suppose  $g \sqsubset_f h$ : Since  $g$  can replace  $h$  there exists a sequence of lattice elements  $x_0(=g), x_1, \dots, x_n(=h)$  such that either  $x_i > x_{i+1}$  and  $\langle x_i, x_{i+1} \rangle \vdash \langle y, f \rangle$  or  $x_{i+1} > x_i$  and  $\langle x_{i+1}, x_i \rangle \vdash \langle f, z \rangle$  for some  $y > f$  and  $f > z$ . It will be shown that  $\forall i$ :  $P_f^*[x_i] \supset P_f^*[x_{i+1}]$  and  $Q_f^*[x_i] \subseteq Q_f^*[x_{i+1}]$  from which the result follows. Suppose  $x_i > x_{i+1}$  then  $P_f^*[x_i] \supset P_f^*[x_{i+1}]$ , so only need to consider the  $Q_f^*$  set. Let  $p \in Q_f^*[x_i]$ , since  $p \in Q_f^*$  there exists  $k \in L$  and an alternating word  $w(x)$  such that  $p > k$  and  $w(p)=y$  and  $w(k)=f$  for some  $y > f$ . Since  $\langle x_i, x_{i+1} \rangle$  is a lower-replaceable-edge it follows that  $x_i \sqsubset_f x_{i+1}$ . Let  $v(x)$  be the alternating word  $v(x)=w((k \vee (p \wedge x)))$ ; then  $v(x_i)=y$ . If  $x_{i+1} \not\geq p$  then  $v(x_{i+1})=f$ , contradicting  $x_i \sqsubset_f x_{i+1}$ , hence  $x_{i+1} \geq p$  and  $Q_f^*[x_i] \subseteq Q_f^*[x_{i+1}]$ .

The case of  $x_{i+1} > x_i$  is dealt with similarly.

$\Leftarrow$ : To show that  $g \sqsubset_f h$  it will suffice to show that there exists a path of upper- and lower-replaceable-edges from  $g$  to  $h$ . Let  $c = g \wedge h$ , and consider the path formed by two chains of elements  $g=x_1, x_2, \dots, x_k=c=y_j, \dots, y_2, y_1=h$  where  $x_i > x_{i+1}$  and  $y_i > y_{i+1}$ . Hence  $P_f^*[h]=P_f^*[c] \subseteq P_f^*[g]$  and  $Q_f^*[g]=Q_f^*[c] \subseteq Q_f^*[h]$ . It will be shown that all edges  $\langle x_i, x_{i+1} \rangle$  are lower-replaceable and all the edges  $\langle y_i, y_{i+1} \rangle$  are upper-replaceable and hence  $g \sqsubset_f h$ .

Suppose for a contradiction that the edge  $\langle y_i, y_{i+1} \rangle$  is a non-upper-replaceable edge. Since  $y_i > y_{i+1}$  there exists a join-irreducible  $p$  such that  $p \leq y_i$  but  $p \not\leq y_{i+1}$ . Let  $n=p \wedge y_{i+1}$  and define  $m$  such that  $p \geq m > n$ . If  $m$  is not a join-irreducible then there exists another join-irreducible  $p'$  such that  $p' \leq m$  but  $p' \not\leq n$ . Since the lattice is finite the above argument can be repeated to show that there exist  $\langle m', n' \rangle$  such that  $m' \leq y_i$  but  $m' \not\leq y_{i+1}$  and  $m'$  is a join-irreducible

and  $y_{i+1} \geq n$ . Since  $m' \vee y_{i+1} = y_i$  and  $n' \vee y_{i+1} = y_{i+1}$  the edge  $\langle m', n' \rangle$  is a non-upper-replaceable and hence  $m' \in P'_f$  and hence  $m' \in P'_f[y_i]$ . However  $m' \notin P'_f[y_{i+1}]$  hence  $m' \notin P'_f[c]$ , contradicting the fact that  $P'_f[y] = P'_f[c]$ .

The proof that every  $\langle x_i, x_{i+1} \rangle$  edge is lower-replaceable is similar. ■

#### §4. The complexity of deciding replaceability and computational equivalence.

In this section, Theorem 2.6 is used as the basis for an algorithm which decides whether two elements  $g, h$  are computationally equivalent or replaceable with respect to a third element  $f$  in a finite lattice  $L$ . It will suffice to describe the following algorithm, which calculates the sets  $P'_f$  and  $\tilde{Q}'_f$  in time polynomial in the size of  $L$ ; to decide whether  $g$  and  $h$  are computationally equivalent or replaceable modulo  $f$  it is only necessary to determine which elements of  $P'_f$  and  $\tilde{Q}'_f$  are less than  $g$  and  $h$ :

##### Algorithm 4.1:

Input: The lattice  $L = \{x_1, \dots, x_N\}$ , together with an  $N \times N$  table specifying all order relations, and an element  $f$  in  $L$ .

Output: The sets  $P'_f$  and  $\tilde{Q}'_f$  which determine the relations  $\sqsubseteq_f$  and  $\sqsupseteq_f$ .

Method:

- 1) Topologically sort the elements of  $L$  into a sequence  $x_1, x_2, \dots, x_N$  such that  $x_i \geq x_j$  implies  $i \geq j$ .
- 2) For each element  $x_i$  scan the elements  $x_1$  to  $x_{i-1}$  from left to right to determine the maximal elements  $< x_i$ . This identifies the subsequence  $p_1, \dots, p_r$  of join-irreducibles (those elements which cover a unique element), and the list of elements which  $x_i$  covers.
- 3) Determine the elements which each  $x_i$  is covered by, and form the subsequence  $q_1, \dots, q_s$  of meet-irreducibles by the dual of Step (2).
- 4) For each join-irreducible  $p_i$  and each meet-irreducible  $q_j$  determine the sets  $\tilde{p}_i$  and  $\tilde{q}_j$  as follows:

```

    for i=1 to r do  $\tilde{p}_i = \varnothing$ .
    for j=1 to s do  $\tilde{q}_j = \varnothing$ .
    for i=1 to r do {
        for j=s downto 1 do {
            if  $q_j \not\leq p_i$  then
                if  $\forall p \in \tilde{q}_j: p_i \not\leq p$  then  $\tilde{q}_j = \tilde{q}_j \cup p_i$ 
                if  $\forall q \in \tilde{p}_i: q_j \not\leq q$  then  $\tilde{p}_i = \tilde{p}_i \cup q_j$ 
            }
        }
    }

```

- 5) For each covering edge  $\langle p, k \rangle$ , where  $p$  is a join-irreducible and  $p \leq f$ , determine whether there is an element  $u$  covered by  $f$  such that  $k \leq u$  and  $p \not\leq u$ ; determine  $P(f)$  as the set of join-irreducibles  $p$  for which such a  $u$  exists. Determine  $Q(f)$  similarly.
- 6) Starting with the set  $P_0 = P(f)$ , repeatedly compute  $P_{i+1} = \tilde{P}_i$  until  $P_k = P_{k+1} = P_f^*$  has been determined. Determine  $\tilde{Q}_f^*$  similarly.

The complexity of steps (1), (2) and (3) is  $O(N^2)$ . Step (4) has complexity  $O(r^3 + s^3)$ . Step (5) has complexity  $O(lr + ms)$  where  $l$  and  $m$  are the number of elements covered and covering  $f$  respectively. Since it is only necessary for an irreducible to appear once in the calculation of  $P_f^*$ , step (6) has complexity  $O(r^3 + s^3)$ . In a lattice (such as  $FDL(t)$ ) where the number of irreducibles and the number of elements around  $f$  is small compared with the size of the lattice, the complexity of this algorithm is  $O(N^2)$ , but it may otherwise be  $O(N^3)$ .

When deciding replaceability and computational equivalence for monotone boolean functions, this algorithm is of course unreasonable, since it requires the order relations in  $FDL(n)$  as input. Indeed:

Theorem 4.2:

The decision problem NONREP:

" Given monotone formulae representing  $f, g, h$  in  $FDL(n)$ , is  $g \not\leq_f h$  ?" is NP-complete.

Proof:

Cor.3.4 in [2] shows that NONREP is in NP: it is only necessary to guess an appropriate prime implicant or prime clause of  $f$  and verify that it distinguishes  $g$  and  $h$ .

Now observe that determining whether a given pair of monotone boolean formula in free variables  $x_1, x_2, \dots, x_n$  represent distinct elements of  $FDL(n)$  is an NP-complete problem. ( If  $w(z_1, z_2, \dots, z_n)$  is a general boolean formula, and  $W(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)$  is a monotone boolean formula, derived from  $w$  by repeated application of de Morgan's laws, such that  $W(z_1, z_2, \dots, z_n, z'_1, z'_2, \dots, z'_n) \equiv w(z_1, z_2, \dots, z_n)$ , then  $w$  is satisfiable if and only if  $(W(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) \vee L) \wedge U \neq L$ , where  $L \equiv \bigvee_{i=1}^n x_i \wedge y_i$  and  $U \equiv \bigwedge_{i=1}^n x_i \vee y_i$ .) Since the monotone boolean formulae  $A$  and  $B$  represent the same element if and only if  $A$  is replaceable modulo  $A$  by  $B$ , it follows that NONREP is an NP-complete decision problem.

A very similar argument can be used to show that non-computational equivalence of monotone boolean functions is also NP-complete.

## §5. Computational equivalence in other contexts.

In this section, replaceability and computational equivalence in particular classes of commutative semigroups is considered.

A semilattice is a commutative semigroup in which every element is idempotent. The semilattice  $(L, \vee)$  is ordered by  $f \leq g$  iff  $f \vee g = g$ .

If  $f, g \in L$ , then the class of  $g$  modulo  $\square_f$  is determined by the set of solutions of the equation  $g \vee x = f$ . This set is empty unless  $g \leq f$ , so that it is essentially enough to consider  $\square_f$  when  $f$  is the largest element of  $L$ .

Theorem 5.1:

Let  $M$  be the set of proper maximal elements of a finite semilattice  $L$ . For  $g$  in  $L$ , let  $M[g]$  denote  $\{m \in M \mid m \geq g\}$ .

Then  $g \sqsubset_1 h$  in  $L$  iff  $M[g] \supset M[h]$ .

Proof: Suppose that  $M[g] \supset M[h]$ . Given  $x$  in  $L$ , we have

$$g \vee x = 1 \text{ if and only if } M[g] \cap M[x] = \emptyset.$$

Thus  $g \vee x = 1$  implies  $h \vee x = 1$ , and  $g \sqsubset_1 h$ .

Conversely, if  $m \in M[h]$  and  $m \notin M[g]$ , then  $m \vee g = 1$ , whilst  $m \vee h \neq 1$ .

Cor. 5.2:

If  $(L, \vee)$  is a finite semilattice, and  $M$  is the set of proper maximal elements of  $L$ , then  $L/\square_1$  is isomorphic with a sub- $\vee$ -semilattice of the boolean lattice  $2^{|M|}$ , and has  $|M|$  proper maximal elements.

Theorem 5.1 applies in particular to a lattice  $L$  which is regarded as a semilattice. If  $D$  is a finite distributive lattice, and  $f \in D$ , then replaceability on the interval  $I = [\lambda(f), \mu(f)]$  has the following curious characterisation:

Cor. 5.3:

$$x \sqsubset_f y \text{ in } (I, \wedge, \vee) \text{ iff } y \sqsubset_{\lambda(f)} x \text{ in } (I, \wedge) \text{ and } y \sqsubset_{\mu(f)} x \text{ in } (I, \vee).$$

Proof:

As described in [1],  $\lambda(f) = \bigvee_{p \in P_f} (p \vee \bar{p})$ , and it is easy to verify that  $\lambda(f)$  is covered by  $|P_f|$  elements, viz. the elements of the form:

$$p' \vee \lambda(f),$$

where  $p' \in P_f$ . If  $y, z \in [\lambda(f), \mu(f)]$ , then

$$y \sqsubset_{\lambda(f)} z \text{ iff } P_f[z] \subseteq P_f[y]$$

by Theorem 5.1, and a dual argument establishes that

$$y \sqsubset_{\mu(f)} z \text{ iff } Q_f[z] \subseteq Q_f[y].$$

The result then follows immediately from [2] Cor. 3.4 (cf Cor. 2.5 above).

It seems to be difficult to deal more generally with replaceability in commutative semigroups; a wide range of different characteristics is observed. For instance, if  $S$  is generated by a single element  $s$  then there is a stem  $k$  and a period  $p$  such that

$$s^k = s^{k+p}$$



and  $S = \{s, s^2, \dots, s^{k-1}, s^k, s^{k+1}, \dots, s^{k+p-1}\}$ . It is easy to verify that if  $f=s^t$ , there are essentially two cases to consider. If  $t < k$ , then  $\square_f$  has the equivalence classes:

$$\{s\}, \{s^2\}, \dots, \{s^{t-1}\}, \{s^t, \dots, s^{k+p-1}\},$$

and  $\square_f$  on  $S/\square_f$  is trivial, whilst if  $t \geq k$ , then  $\square_f$  is defined by

$$s^{t+i} \square_f s^{t+i-p}$$

for  $1 \leq i \leq k+p-t-1$ , so that  $S/\square_f$  has stem  $k' \leq k$  and period  $p$ , and  $\square_f$  on  $S/\square_f$  is defined by  $s^i \square_f s^{i+p}$  for  $1 \leq i < k'$ .

As a less trivial example, computational equivalence in  $(\mathbb{Z}_n, \cdot)$  is characterised using the propositions below, which justify the following procedure for determining  $(\mathbb{Z}_n/\square_f, \square_f, \cdot)$  for each residue  $f$  such that  $0 \leq f < n$ :

1. let  $d=(f, n)$ , where  $d=n$  if  $f=0$
2. partition the group of units  $\mathbb{Z}_n^*$  of  $\mathbb{Z}_n$  (the residues coprime to  $n$ ) into congruence classes  $U_1, U_2, \dots, U_t$  modulo  $n/d$ .
3. construct the equivalence classes of  $\square_f$  as the sets of the form:

$$rU_i \text{ where } r \mid d, \text{ and } 1 \leq i \leq t,$$

together with  $Z = \mathbb{Z}_n \setminus \bigcup_{r \mid d} \bigcup_{i=1}^t rU_i$  if  $Z$  is non-empty (that is, if  $f \neq 0$ ).

4. note that for any equivalence class  $C \neq Z$ , the residue class of  $x$  modulo  $n/d$ , and  $(x, n)$  are both independent of the choice of  $x$  within  $C$ ; denoting these invariants by  $\rho(C)$  and  $\delta(C)$  respectively, construct  $\square_f$  as the set of replacements of the form  $C \square_f C'$ , for which  $C=Z$ , or neither  $C$  nor  $C'$  is  $Z$  and  $\exists$  equivalence classes  $A, B$  and  $B'$  such that  $C=A.B$ ,  $C'=A.B'$ ,  $\rho(B)=\rho(B')=1$  and  $\delta(B) \mid \delta(B')$ .

This procedure is based upon the characterisation of  $\square_f$  and  $\square_f$  given in Theorem 5.6, Cor. 5.8 and Theorem 5.9 below; some simple preliminary lemmas are required.

Lemma 5.4: If  $g < n$  and  $(g, n)=r$ , then  $\exists a \in \mathbb{Z}_n^*$  such that  $ag \equiv r \pmod{n}$ .

Proof:

$(g, n)=r < n$ , whence  $\exists k$  such that  $1 \leq k < n/r$  and  $g=kr$ . Moreover  $(k, n/r)=1$ .

Let  $p_1, \dots, p_t$  be the prime factors of  $n$  which do not divide  $k$ . (This is a non-empty set, since  $(k, n/r)=1$  and  $r < n$ .) Then  $a=p_1 \cdots p_t(n/r)+k$  is co-prime to  $n$ : if  $p \mid n$ , then

either  $p \mid k$  and  $p \nmid p_1 \cdots p_t(n/r)$

or  $p \nmid k$  and  $p \mid p_1 \cdots p_t(n/r)$ .

Moreover,  $g = kr \equiv ar \pmod{n}$ , whence  $r \equiv \bar{a}g \pmod{n}$ .

Lemma 5.5: If  $a \in \mathbb{Z}_n^*$  and  $f \in \mathbb{Z}_n$ , then

(i)  $\square_f = \square_{af}$

(ii)  $g \square_f h$  iff  $ag \square_f ah$

Proof:

(i) Suppose that  $g \square_f h$ . Then

$$gx = af \Rightarrow g\bar{a}x = f \Rightarrow h\bar{a}x = f \Rightarrow hx = af,$$



so that  $g \sqsubset_{af} h$ . Similarly  $g \sqsubset_{af} h \Rightarrow g \sqsubset_{\bar{a}(af)=f} h$ .

(ii) Suppose that  $g \sqsubset_f h$ . Then

$$agx = f \Rightarrow gx = \bar{a}f \Rightarrow hx = \bar{a}f \text{ by (i),}$$

from which it follows that  $ahx = f$ . This proves that  $ag \sqsubset_f ah$ , and the converse follows by symmetry. ■

Lemma 5.4 and Lemma 5.5(i) show that it is sufficient to consider the relations  $\sqsubset_d$  where  $d \mid n$ . Indeed,  $\sqsubset_f$  coincides with  $\sqsubset_d$ , where  $d=(f,n)$ . (The special case  $f=0$  is obtained by taking  $d=0$ .)

Consider then the computational equivalence class of  $g$  modulo  $d$ , where  $d$  is a divisor of  $n$ , and  $(g,n)=r$ . If  $d < n$  and  $r \nmid d$ , the congruence  $gx \equiv d \pmod{n}$  is insoluble; in this case,  $g \sqsubset_d 0$ , and  $g$  is replaceable by any other residue when computing  $d$ . Otherwise:

Theorem 5.6: If  $(g,n)=r \mid d \mid n$ , then

$$g \sqsubset_d h \text{ iff } h \equiv kg \pmod{n} \text{ where } k \equiv 1 \pmod{n/d}.$$

Proof:

Suppose that  $h \equiv kg \pmod{n}$ , where  $k \equiv 1 \pmod{n/d}$ . Then

$$gx \equiv d \pmod{n} \Rightarrow hx \equiv kgx \equiv kd \equiv d \pmod{n}.$$

Conversely, suppose that  $g \sqsubset_d h$ , and let  $a \in \mathbb{Z}_n^*$  be chosen as in Lemma 5.4 so that  $ag \equiv r \pmod{n}$ . From  $r \sqsubset_d \bar{a}h$  and  $r \cdot (d/r) = d$ , it follows that  $\bar{a}h \cdot (d/r) \equiv d \pmod{n}$ , and that  $ah$  is a multiple of  $r$ , say  $kr$ . Moreover,  $kd \equiv d \pmod{n}$ , so that  $k \equiv 1 \pmod{n/d}$ , and

$$h \equiv a\bar{a}h \equiv k\bar{a}r \equiv kg \pmod{n}.$$

Cor. 5.7:

If  $(g,n)=r \mid d \mid n$ , then

$$g \sqsubset_d h \Rightarrow (g,n) \mid (h,n) \mid d \text{ and } (gd,n) = (hd,n).$$

Proof:

Suppose that  $(g,n)=r \mid d \mid n$ , and that  $g \sqsubset_d h$ . Since the congruence  $gx \equiv d \pmod{n}$  is soluble by Lemma 5.4, so also is  $hx \equiv d \pmod{n}$ , proving that  $(h,n) \mid d$ . By Theorem 5.6, there is a  $k \equiv 1 \pmod{n/d}$  such that  $h \equiv kg \pmod{n}$ . From this congruence, it follows that  $r \mid (h,n)$ . Moreover:

$$hd = kgd = gd \pmod{n}$$

proving that  $(hd,n) = (gd,n)$ . ■

Cor. 5.8: If  $(g,n) \mid d \mid n$ , and  $g \equiv h \equiv 1 \pmod{n/d}$  then

$$g \sqsubset_d h \text{ iff } (g,n) \mid (h,n)$$

Proof:

If  $(g,n) \mid (h,n)$ , then  $\exists k$  such that  $g \equiv kh \pmod{n}$ , by Lemma 5.4. Certainly  $k \equiv 1 \pmod{n/d}$ , since  $g \equiv h \equiv 1 \pmod{n/d}$ , so that  $g \sqsubset_d h$  by Theorem 5.6.

The converse follows from Cor. 5.7. ■

Theorem 5.9: If  $g, h \in \mathbf{Z}_n$ , and  $(g, n) = r \mid d \mid n$ , then

$$g \sqsubseteq_d h \text{ iff } (h, n) = r \text{ and } \exists a, b \in \mathbf{Z}_n^* \text{ such that} \\ a \equiv b \pmod{n/d} \text{ and } g \equiv ar, h \equiv br \pmod{n}$$

Proof: Suppose that  $(h, n) = r$ , and that  $a, b \in \mathbf{Z}_n^*$  satisfy

$$g \equiv ar \pmod{n}, h \equiv br \pmod{n} \text{ and } a \equiv b \pmod{n/d}.$$

Then  $g \equiv kh \pmod{n}$ , where  $k \equiv a\bar{b} \pmod{n}$ , and  $k \equiv 1 \pmod{n/d}$  since  $a \equiv b \pmod{n/d}$ . Thus  $g \sqsubseteq_d h$  by Theorem 5.6, and  $h \sqsubseteq_d g$  by symmetry.

For the converse: suppose that  $g \sqsubseteq_d h$ . Cor. 5.7 establishes directly that  $(g, n) = (h, n) = r$ . Applying Lemma 5.4, let  $a, b \in \mathbf{Z}_n^*$  be such that  $ag \equiv bh \equiv r \pmod{n}$ , and choose  $x$  such that  $gx \equiv d \pmod{n}$ . Then

$$\bar{a}rx \equiv gx \equiv d \pmod{n} \Rightarrow hx \equiv \bar{b}rx \equiv d \pmod{n},$$

whence  $ad \equiv rx \equiv bd \pmod{n}$ , proving that  $a \equiv b \pmod{n/d}$ . ■

Cor. 5.10: Suppose that  $(g, n) = r \mid d \mid n$ . If  $g \sqsubseteq_d h$  and  $(h, n) = r$ , then  $g \sqsubseteq_d h$ .

Proof:

By Theorem 5.6,  $\exists k$  such that  $h = kg$ , and  $k \equiv 1 \pmod{n/d}$ . By Lemma 5.4,  $\exists u \in \mathbf{Z}_n^*$  such that  $h \equiv ug \pmod{n}$ . Thus  $kg \equiv ug \pmod{n}$ , proving that  $k \equiv u \pmod{n/r}$ , and that  $k \equiv u \equiv 1 \pmod{n/d}$ . Theorem 5.9 then shows that  $h \sqsubseteq_d g$ . ■

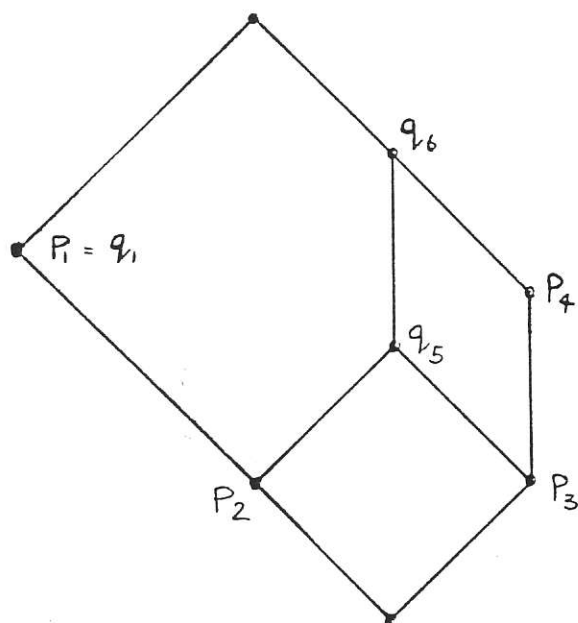
## References.

1. Beynon, W.M.: Replacement in monotone boolean networks: an algebraic perspective. In: Proc. 4th. FST&TCS, Bangalore, Lecture Notes in Computer Science 181, Springer-Verlag, Heidelberg 1984.
2. Beynon, W.M.: Replaceability and computational equivalence for monotone boolean functions. Acta Informatica, 1985 (to appear).
3. Beynon, W.M.: Monotone boolean functions computable by planar circuits. Univ. of Warwick, Theory of Computation Report 67, 1984.
4. Buckle, J.F.: A characterisation of meet and join respecting preorders and congruences on a finite lattice. (unpublished manuscript).
5. Dilworth, R.P.: Algebraic theory of lattices. Prentice-Hall, 1973.
6. Dilworth, R.P.: Structure of relatively complemented lattices, Annals of Maths. 51, 348-359, 1950.
7. Dunne, P.E.: Techniques for the analysis of monotone boolean networks. Univ. of Warwick, Theory of Computation Report 69, 1984.
8. Grätzer, G.: Lattice Theory: first concepts and distributive lattices. San Francisco: W.H. Freeman and Co. 1971.
9. Jürgensen, H. and Thierrin, G.: Semigroups with each element disjunctive. Semigroup Forum Vol.21, 127-141, 1980.

10. Lallement, G.: Semigroups and combinatorial applications, Wiley, NY, 1979.
11. Mehlhorn, K. & Galil, Z.: Monotone switching networks and boolean matrix product. Computing (16), 99-111, 1976.
12. Paterson, M.S.: Complexity of monotone networks for boolean matrix product. Theoretical Computer Science (1), 13-20, 1975.
13. Shyr, H.J.: Free monoids and languages. Lecture Notes, Dept. of Maths, Soochow Univ., Taipei, Taiwan, 1979.

### Some illustrative examples.

The simple examples below serve to complement the description of computational equivalence and replaceability in finite lattices given above. Example 1 shows that Proposition 2.2 is false for general lattices. Example 2 illustrates the definition of the set  $P_f$ . Example 3 illustrates the construction of the sets  $P_f^*$  and  $\tilde{Q}_f^*$ , and the form of the computational equivalence and replaceability relations. Example 4 shows that computational equivalence is not generally associated with a retract onto an interval.



$$\mathcal{P}_4 = \{ q_1, q_5 \}$$

$$\mathcal{Q}_1 = \{ p_3 \}$$

$$\langle p_1, p_2 \rangle \rightarrow \langle p_4, p_3 \rangle$$

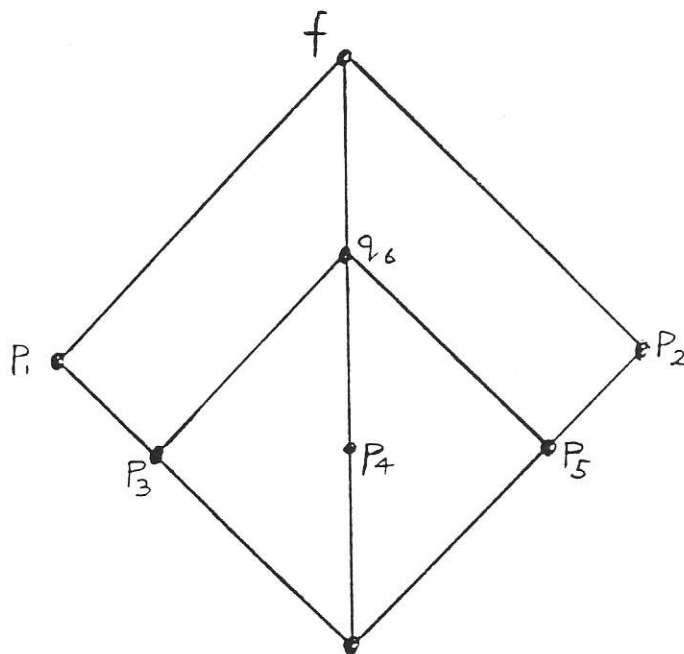
$$\langle p_4, p_3 \rangle \vdash \langle p_1, p_2 \rangle$$

Example 1.

$$P(f) = \{ p_1, p_2, p_3, p_4, p_5 \}$$

$$P_f = \{ p_1, p_4, p_5 \}$$

Example 2.



Example 3.

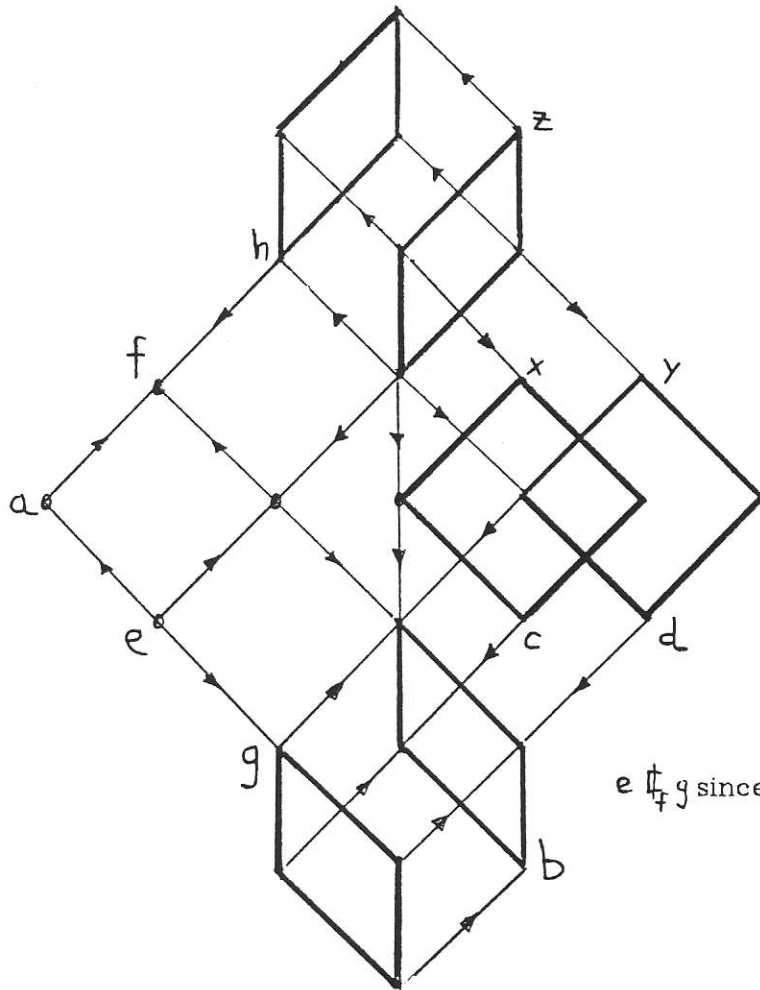
$$P_f = \{a, b\} \quad Q_f = \{f\}$$

$$\tilde{P}_f = \{z, a\} \quad \tilde{Q}_f = \{c, d\}$$

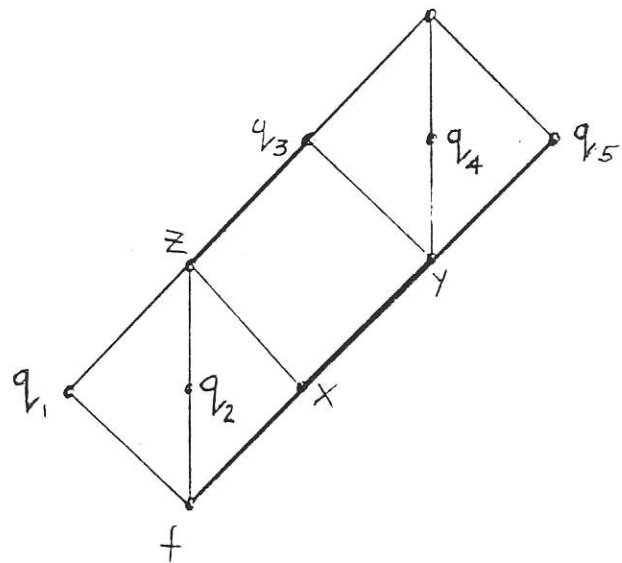
$$\tilde{P}_f = \{a, b\} \quad \tilde{Q}_f = \{f, y, x\}$$

$$\tilde{Q}_f^s = \{c, d, e\}$$

Bold lines join computationally equivalent elements. Arrows point to computationally more powerful elements.



$$e \not\vdash_T g \text{ since } (((e \wedge e) \vee x) \wedge d) \vee f = h, (((g \wedge e) \vee x) \wedge d) \vee f = f$$



$$\tilde{Q}_f^* = \{q_1, q_2, q_4, q_5, x\}$$

$$z \sqsubset_T q_5 \text{ \& } x \sqsubset_T y$$

Example 4.

The characterisation of computational equivalence and replaceability in  $(\mathbb{Z}_n, \cdot)$  given in §5 is illustrated by the following example, which describes these relations for  $n=12$ :

The units of  $(\mathbb{Z}_{12}, \cdot)$  are 1,5,7,11. The possible computational equivalence relations are determined by the divisors of 12, viz. 1,2,3,4,6,12. The relations  $\sqsubset_d$  and  $\sqsupset_d$  for each divisor  $d$  of 12 are as follows:

$d=1$ : The computational equivalence classes are  $\{1\} \{5\} \{7\} \{11\} \{0\} \{2\} \{3\} \{4\} \{6\} \{8\} \{9\} \{10\}$ , and there are no non-trivial replaceability relations.

$d=2$ : The units separate into two classes modulo  $n/d=6$ , viz.  $\{1\} \{7\} \{5\} \{11\}$ . The other computational equivalence classes are  $\{2\} \{10\}$  and  $\{0\} \{3\} \{4\} \{6\} \{8\} \{9\}$ . There are no non-trivial replaceability relations.

$d=3$ : The units separate into two classes modulo  $n/d=4$ , viz.  $\{1\} \{5\} \{7\} \{11\}$ . The other computational equivalence classes are  $\{3\} \{9\}$  and  $\{0\} \{2\} \{4\} \{6\} \{8\} \{10\}$ . The only non-trivial replaceability relation for elements  $\equiv 1 \pmod{n/d=4}$  is  $\{1\} \{5\} \sqsupset_d \{9\}$ . The only other non-trivial replacement is  $\{7\} \{11\} \sqsupset_d \{3\}$ . There are no non-trivial replaceability relations.

$d=4$ : The units separate into two classes modulo  $n/d=3$ , viz.  $\{1\} \{7\} \{5\} \{11\}$ . The other computational equivalence classes are  $\{4\} \{8\} \{2\} \{10\}$  and  $\{0\} \{3\} \{6\} \{9\}$ . The non-trivial replaceability relations for elements  $\equiv 1 \pmod{n/d=3}$  are

$$\{1\} \{7\} \sqsupset_d \{10\} \sqsupset_d \{4\}.$$

The only other non-trivial replacements are  $\{5\} \{11\} \sqsupset_d \{2\} \sqsupset_d \{8\}$ .

$d=6$ : The units define a single class modulo  $n/d=2$ , viz.  $\{1\} \{5\} \{7\} \{11\}$ . The other computational equivalence classes are  $\{2\} \{10\} \{3\} \{9\} \{6\}$  and  $\{0\} \{4\} \{8\}$ . The only non-trivial replaceability relation for elements  $\equiv 1 \pmod{n/d=2}$  is

$$\{1\} \{5\} \{7\} \{11\} \sqsupset_d \{3\} \{9\}.$$

The only other non-trivial replacement is  $\{2\} \{10\} \sqsupset_d \{6\}$ .

$d=12$ : The units define a single class modulo  $n/d=1$ , viz.  $\{1\} \{5\} \{7\} \{11\}$ . The other computational equivalence classes are  $\{2\} \{10\} \{3\} \{9\} \{4\} \{8\} \{6\}$  and  $\{0\}$ . In this case,  $n/d=1$ , so that all residues are  $\equiv 1 \pmod{n/d}$ , and the valid replacement relations are defined by the divisors of 12 ordered by divisibility: viz.

$$\{1\} \{5\} \{7\} \{11\} \sqsupset_d \{2\} \{10\} \sqsupset_d \{6\} \sqsupset_d \{0\}, \{1\} \{5\} \{7\} \{11\} \sqsupset_d \{3\} \{9\} \sqsupset_d \{6\}, \{2\} \{10\} \sqsupset_d \{4\} \{8\} \sqsupset_d \{0\}.$$

That is, the ordering by replaceability defines the ideal lattice of the ring  $(\mathbb{Z}_n, \cdot)$ .